

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of :  
Masaki TOYOKURA : **Confirmation No. 5157**  
Serial No. 10/802,933 : **Attn: BOX MISSING PARTS**  
Filed March 18, 2004 : **Attorney Docket No.2004\_0444A**  
BUS CONTROLLER :

**CLAIM OF PRIORITY UNDER 35 USC 119**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

Applicant in the above-entitled application hereby claims the date of priority under the International Convention of Japanese Patent Application No. 2003-076055, filed March 19, 2003, as acknowledged in the Declaration of this application.

A certified copy of said Japanese Patent Application is submitted herewith.

Respectfully submitted,

Masaki TOYOKURA

THE COMMISSIONER IS AUTHORIZED  
TO CHARGE ANY DEFICIENCY IN THE  
FEES FOR THIS PAPER TO DEPOSIT  
ACCOUNT NO. 23-0975

By

Jeffrey R. Filipek  
Registration No. 41,471  
Attorney for Applicant

JRF/fs  
Washington, D.C. 20006-1021  
Telephone (202) 721-8200  
Facsimile (202) 721-8250  
June 25, 2004

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日            2 0 0 3 年   3 月 1 9 日  
Date of Application:

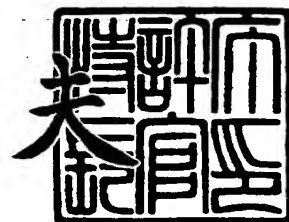
出 願 番 号            特 願 2 0 0 3 - 0 7 6 0 5 5  
Application Number:  
[ST. 10/C]:            [ J P 2 0 0 3 - 0 7 6 0 5 5 ]

出 願 人            松下電器産業株式会社  
Applicant(s):

2 0 0 3 年   9 月 1 0 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康



出証番号   出証特 2 0 0 3 - 3 0 7 4 2 6 2

【書類名】 特許願

【整理番号】 2037840149

【提出日】 平成15年 3月19日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 13/362  
G06F 12/00  
G06F 13/18

【発明者】

【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内

【氏名】 豊蔵 真木

【特許出願人】

【識別番号】 000005821

【氏名又は名称】 松下電器産業株式会社

【代理人】

【識別番号】 100081813

【弁理士】

【氏名又は名称】 早瀬 憲一

【電話番号】 06(6395)3251

【手数料の表示】

【予納台帳番号】 013527

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9600402

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 バスコントローラ

【特許請求の範囲】

【請求項 1】 共有メモリを使用する複数のリクエストの処理レベルを制御するバスコントローラであって、

上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、

上記各リクエスト毎に異なる処理レベルの処理を行う処理手段と、

上記各リクエスト毎に異なる処理レベルの処理と上記アクセスサイクル数との対応を記述した対応表と、

アクセスが許可される各リクエストの処理手段による処理レベルを指定する処理レベル判定手段と、を備え、

上記処理レベル判定手段は、上記アクセスサイクル数カウンタによりカウントされる現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記各リクエスト毎に異なる処理レベルの処理と上記アクセスサイクル数との対応表とに従い、アクセスが許可される各リクエストの処理レベルを指定する、ことを特徴とするバスコントローラ。

【請求項 2】 共有メモリに対してアクセスする複数のリクエストからのアクセス要求に対応して調停するバスコントローラであって、

上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、

上記複数のリクエストとアクセスサイクル数との対応表と、

上記複数のリクエストからのアクセス要求に対してアクセスの許可を調停するアービタと、を備え、

上記アクセスサイクル数カウンタにより上記共有メモリをアクセスする現サイクル数をカウントし、

上記アービタは、上記アクセスサイクル数カウンタによりカウントされる現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記複数のリクエストとアクセスサイクル数との対応表とに従い、全リクエストのサイク

ル数の合計が、限界サイクル数を越えることが予測される場合には、非リアルタイム系のバスアクセス要求を許可しない制御をする、  
ことを特徴とするバスコントローラ。

【請求項 3】 共有メモリを使用する複数のリクエストの処理レベルを制御し、複数のリクエストからのアクセス要求に対応して調停するバスコントローラであって、

上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、

上記各リクエスト毎に異なる処理レベルの処理を行う処理手段と、

上記各リクエスト毎に異なる処理レベルの処理とアクセスサイクル数との対応を記述した対応表と、

アクセスが許可される各リクエストの処理手段による処理レベルを指定する処理レベル判定手段と、

上記複数のリクエストからのアクセス要求に対応してアクセスの許可を調停するアービタと、を備え、

上記処理レベル判定手段、および上記アービタは、上記アクセスサイクル数カウンタによりカウントされる現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記対応表とに従い、アクセスが許可されるリクエストの処理手段による処理レベルを指定するとともに、全リクエストのサイクル数の合計が、限界サイクル数を越えることが予測される場合には、非リアルタイム系のバスアクセス要求を許可しない制御をする、

ことを特徴とするバスコントローラ。

【請求項 4】 請求項 1 記載のバスコントローラにおいて、

上記処理レベル判定手段は、リクエストの 2 個後の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、

限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、

上記残サイクル数から上記総和を減算したサイクル数までの範囲で次のリクエ

スタの処理のレベルを選択する、

ことを特徴とするバスコントローラ。

【請求項 5】 請求項 2 記載のバスコントローラにおいて、

上記アービタは、次のリクエストである非リアルタイム系リクエストの次の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、

限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、

上記残サイクル数から上記総和を減算したサイクル数までの範囲で処理を完了できない場合は次の非リアルタイム系のリクエストを許可しない、

ことを特徴とするバスコントローラ。

【請求項 6】 請求項 3 記載のバスコントローラにおいて、

上記処理レベル判定手段、および上記アービタは、次のリクエストである非リアルタイム系リクエストの次の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、

限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、

上記残サイクル数から上記総和を減算したサイクル数までの範囲で処理を完了できない場合は、次の非リアルタイム系のリクエストを許可しない、

ことを特徴とするバスコントローラ。

【請求項 7】 請求項 6 記載のバスコントローラにおいて、

上記非リアルタイム系リクエストは、複数の異なる処理レベルを有するものである、

ことを特徴とするバスコントローラ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、バスコントローラに関し、特に、AVデータ、及び通信情報を処理

するシステムにおける共有メモリのバスコントローラに関する。

#### 【0002】

##### 【従来の技術】

A V データを、通信または記録再生する A V 処理システムにおいて、通信手段の帯域幅や、記録媒体の容量を削減するために、画像や音声を符号化し、再生時には復号化する、ことが行われている。国際標準の動画像の符号化方式として、M P E G 1 / 2 / 4 などがある。これらの符号化方式は、動きベクトル検出、及び動き補償、D C T (Discrete Cosine Transform)、量子化、V L C (Variable Length Coding)などを組合せた方法である。また、高画質化や、画像の縮小、拡大時には、フィルタリング、さらに、文字データ処理や、グラフィックス処理、通信経路からのデータ転送、などの処理を行なう必要がある。このような A V 処理システムでは、各処理を行なうリクエストが、共有メモリを用いて各々の処理を行なっている。

#### 【0003】

従来のバスコントローラとしては、例えば、アービタの優先順位変更を行なう特許文献のようなものがある。この従来例を単純化して示した図 6 により、従来技術の説明を行なう。図 6 において、6 1 は共有メモリ、6 2 はバスコントローラ、6 3 は第 1 のリクエスト、6 4 は第 2 のリクエスト、6 5 は第 3 のリクエストである。

#### 【0004】

A V 処理システムの例として、画像符号化システムを考え、第 1 のリクエスト 6 3 として、画像のフィルタリング処理を、第 2 のリクエスト 6 4 として、画像符号化処理を、第 3 のリクエスト 6 5 として、情報通信インタフェースによるデータ転送処理を、考えるものとする。

#### 【0005】

第 1 のリクエスト 6 3 と、第 2 のリクエスト 6 4 は、リアルタイム系の処理であり、画像 1 フレームの時間に定まった量の共有メモリへのアクセス、及び、データ処理を必要とする。これに対して、第 3 のリクエスト 6 5 は、非リアルタイム系の処理であり、画像の処理に比較して、非定期的な共有メモリへのアクセス



、及び、データ処理を行なう。

#### 【0006】

バスコントローラ62の構成例を、図5に示す。図5において、51はアービタ、52はプロトコル変換手段である。各リクエストからの要求信号に従い、アービタ51が優先順位に従って、上記リクエストのうちの一つに許可信号（共有メモリアクセス信号（CMA））を与える。また、プロトコル変換手段52は、リクエストのアクセスに用いられるアドレス、データ、及び、READかWRITEを示すリクエストアクセス信号（RA）を、共有メモリアクセス信号（CMA）にプロトコル変換する機能を持つとともに、該プロトコル変換や、アクセス速度の変換のためのバッファ機能を持つ。共有メモリアクセス信号（CMA）としては、SDRAMアクセスのためのRowアドレス、Columnアドレス出力、及び、コマンド発行、などがある。

#### 【0007】

第1リクエスト63、第2リクエスト64、第3のリクエスト65は、各リクエスト内の処理に従い、バスコントローラ62に共有メモリ61へのアクセス要求を出し、バスコントローラ62の許可に従って、バスコントローラ62のプロトコル変換を通して、共有メモリ61にアクセスを行なう、すなわち、共有メモリアクセス信号（CMA）により、アクセスする。

#### 【0008】

図7に、サイクル数の例を示す。図7（a）に、各リクエストのサイクル数の例を示し、簡単のため、第1のリクエスト（REQ1）、第2のリクエスト（REQ2）は、1フレーム時間内に、2回ずつのアクセスを必要とし、第3のリクエスト（REQ3）は、1フレーム時間内に、0回または1回のアクセスを必要とし、各リクエストは、定まったワード数のデータのアクセスを必要としている。サイクル数は、共有メモリ側のクロック数とする。各リクエストのサイクル数でアクセスするアドレスによって、共有メモリ側のクロック数が変化することがある。

#### 【0009】

SDRAMの場合、同じワード数のデータをアクセスする場合でも、ロード

レスが変更、連続アドレスかどうかに従い、最も少ないサイクル数でアクセスするためには、SDRAMへ必要なアドレス情報や、コマンド、シーケンスに変更することが必要となる。

#### 【0010】

第1のリクエスト (REQ1) で、最大のサイクル数を、 $N1(max)$ 、最も少ないサイクル数を、 $N1(min)$  とし、平均的なサイクル数を、 $N1(ave.)$  として示し、 $N1(max) - N1(ave.)$  を黒色で示し、 $N1(ave.) - N1(min)$  を白色で示している。第2のリクエスト (REQ2) についても、同様である。

#### 【0011】

図7(b)で、1フレーム内の時間での全サイクル数と、そのサイクル数の設計について示している。1フレーム時間内の共有メモリアクセスのサイクル数は、

$$N1(max) \times 2 + N2(max) \times 2 + N3(max) \times 1$$

の場合が最大となる。この場合がありうるので、設計上システムが破綻しないように、このサイクル数を1フレーム時間に確保することが必要となる。ただし、これはすべてのアクセスが最大サイクル数になる場合であるので、頻度としては少ないものとなることが予想される。

#### 【0012】

典型的な場合として、すでに示したように、1フレーム時間内に第3のリクエストがなく、 $N1(ave.) \times 2 + N2(ave.) \times 2$  の場合、 $N1(min) + N2(max) + N1(max) + N2(min)$  の場合などがあり、このような場合が、比較的頻度が多くなると考えられる。また、最小は1フレーム時間内に、 $N1(min) \times 2 + N2(min) \times 2$  で済む。

#### 【0013】

##### 【特許文献1】

特開 2001-184300号 (第5頁0026～第6頁0047、  
第1図)

#### 【0014】

**【発明が解決しようとする課題】**

上記の従来例では、頻度は少なくとも起こりうる最大のサイクル数で設計するため、1フレーム時間当たりのサイクル数が多く、共有メモリの動作周波数が高くなり、共有メモリや、回路のコストアップや、設計難易度が高くなる、ということが起こりうる。

**【0015】**

また、システム設計後にアプリケーションの改変、増加により、共有メモリアクセスのサイクル数が増加した場合には、動作周波数を上げた新しいバスシステムが必要となり、設計の難易度と、再設計による設計、製造のコストアップに繋がることとなる。

**【0016】**

本発明は上記従来の問題点を解決するためになされたもので、頻度が少ないような最大サイクル数に対応する高い周波数によるバスシステム設計によるコストアップを抑え、最大サイクル数が増加した場合でも、バスシステムの再設計を抑えるために最大サイクル数よりも小さいサイクル数、すなわち低い周波数で設計することのできるバスコントローラを提供することを目的としている。

**【0017】****【課題を解決するための手段】**

上記課題を解決するために、本発明の請求項1にかかるバスコントローラは、共有メモリを使用する複数のリクエストの処理レベルを制御するバスコントローラであって、上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、上記リクエスト毎に異なる処理レベルの処理を行う処理手段と、上記複数のリクエストの異なる処理レベルの処理と、上記アクセスサイクル数との対応を記述した対応表と、上記処理手段による処理レベルを指定する処理レベル判定手段と、を備え、上記アクセスサイクル数カウンタにより上記共有メモリをアクセスする現サイクル数をカウントし、上記処理レベル判定手段により上記現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記対応表とに従い、各々のリクエストの処理レベルを指定するものである。

## 【0018】

また、本発明の請求項2にかかるバスコントローラは、共有メモリを使用する複数のリクエストからのアクセス要求に対応して調停するバスコントローラであって、上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、上記複数のリクエストの処理と、アクセスサイクル数との対応表と、上記複数のリクエストからのアクセス要求に対応してアクセスの許可を調停するアービタと、を備え、上記アクセスサイクル数カウンタにより上記共有メモリをアクセスする現サイクル数をカウントし、上記アービタにより上記現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記対応表とに従い、限界サイクル数を越えることが予測される場合には非リアルタイム系のバスアクセス要求を許可しない制御をするものである。

## 【0019】

また、本発明の請求項3にかかるバスコントローラは、共有メモリを使用する複数のリクエストの処理レベルを制御し、複数のリクエストからのアクセス要求に対応して調停するバスコントローラであって、上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、上記リクエスト毎に異なる処理レベルの処理を行う処理手段と、上記複数のリクエストの異なる処理レベルの処理と、アクセスサイクル数との対応を記述した対応表と、上記処理手段による処理レベルを判定する処理レベル判定手段と、上記複数のリクエストからのアクセス要求に対応してアクセスの許可を調停するアービタと、を備え、上記アクセスサイクル数カウンタにより上記共有メモリをアクセスする現サイクル数をカウントし、上記処理レベル判定手段、および上記アービタにより上記現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記対応表とに従い、各々のリクエストの処理レベルを指定するとともに、限界サイクル数を越えることが予測される場合には非リアルタイム系のバスアクセス要求を許可しない制御をするものである。

## 【0020】

また、本発明の請求項4にかかるバスコントローラは、請求項1記載のバスコントローラにおいて、上記処理レベル判定手段は、2個後の、上記リクエストに

対する処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で次のリクエストの処理のレベルを選択するものである。

#### 【0021】

また、本発明の請求項5にかかるバスコントローラは、請求項2記載のバスコントローラにおいて、上記アービタは、非リアルタイム系リクエストの次の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で処理を完了できない場合は次の非リアルタイム系のリクエストを許可しないものである。

#### 【0022】

また、本発明の請求項6にかかるバスコントローラは、請求項3記載のバスコントローラにおいて、上記処理レベル判定手段は、次の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で処理を完了できない場合は次の非リアルタイム系のリクエストを許可しないものである。

#### 【0023】

また、本発明の請求項7にかかるバスコントローラは、請求項6記載のバスコントローラにおいて、上記非リアルタイム系リクエストは、複数の、異なるレベルの処理レベルを有するものである。

#### 【0024】

これらの手段により、全リクエストについての最大アクセスサイクル数と最大アクセス回数の総和よりも小さいサイクル数で、設計することが可能となる。

#### 【0025】

**【発明の実施の形態】****(実施の形態 1)**

図 1 は本発明の実施の形態 1 によるバスコントローラの構成図、図 2 は本実施の形態 1 におけるバスシステムの構成例を示す図、図 3 は本実施の形態 1 における第 1 の処理サイクルの説明図、図 4 は本実施の形態 1 における第 2 の処理サイクルの説明図である。

**【0026】**

図 2 において、21 は共有メモリ、22 はバスコントローラ、23 は第 1 のリクエスタ、24 は第 2 のリクエスタである。

AV 処理システムの例として、画像符号化システムを考え、第 1 のリクエスタ 23 として、画像のフィルタリング処理をするもの、第 2 のリクエスタ 24 として、画像符号化処理をするもの、を考える。第 1 のリクエスタ 23 と、第 2 のリクエスタ 24 は、リアルタイム系の処理であり、画像 1 フレームの時間にそれぞれ 2 種類の処理レベルを持ち、それぞれの処理レベルに従った定まった量の共有メモリへのアクセス、及びデータ処理、を必要とする。2 種類の処理レベルで、共有メモリのアクセスサイクル数が多い方を A レベル、少ない方を B レベルとする。第 1 のリクエスタ 23 は、画像のフィルタリング処理であり、A レベルとして、現フレームと前フレームのデータを用いたフィルタリング処理を、B レベルとして、現フレームのデータを用いたフィルタリング処理を、行うものとする。第 2 のリクエスタ 24 は、画像符号化処理であり、A レベルとして、広い動きベクトル検出範囲用の参照画像を用いた画像符号化処理を、B レベルとして、狭い動きベクトル検出範囲用の参照画像を用いた画像符号化処理を、行うものとする。両方のリクエスタとも、A レベルの方が、画質は良くなるが、メモリサイクル数は多くなるものとする。

**【0027】**

バスコントローラ 22 の構成例を、図 1 に示す。図 1 において、11 はアービタ、12 はサイクルカウンタ、13 はリクエスタ処理レベルーメモリサイクル対応表、14 はプロトコル変換手段である。各リクエスタからの要求信号に従い、アービタ 11 が、優先順位に従ってリクエスタのうちの一個に許可信号と、処理

レベル制御信号とを与える。第1のリクエスト、第2のリクエストの順の繰り返してアクセス要求が入力され、これらに対して順に許可を与える。

#### 【0028】

サイクルカウンタ12は、メモリプロトコル変換手段14からの情報に従い、共有メモリのアクセスサイクル数をカウントし、アクセス毎に現時点のサイクル数を出力する。

#### 【0029】

アービタ11は、現時点のサイクル数と、予め定められた限界サイクル数から残りサイクル数を計算し、リクエスト処理レベルーサイクル数対応表13の情報を用いて、共有メモリ21へのアクセスを次にどのリクエストに許可するかを決定し、調停信号であるアービタ制御信号ACを、出力するとともに、該アクセスを許可したリクエストの処理レベルを決定して、処理レベル制御信号PLとして出力する。

#### 【0030】

メモリプロトコル変換手段14は、リクエストのアクセスに用いられるアドレス、データ、READかWRITEを示すリクエストアクセス信号RAを、共有メモリアクセス信号CMAにプロトコル変換するとともに、プロトコル変換やアクセス速度の変換のためのバッファ機能を持つものである。共有メモリアクセス信号CMAとしては、SDRAMアクセスのためのRowアドレス、Columnアドレス、コマンド発行などがある。

#### 【0031】

第1リクエスト23、第2リクエスト24は、各リクエスト内の処理に従い、バスコントローラ22に共有メモリ21へのアクセス要求を出し、バスコントローラ22の許可と、処理レベル制御信号とに従って、バスコントローラ22のプロトコル変換を通して、共有メモリ21にアクセスを行なう。

#### 【0032】

図3のサイクル数の例で動作を詳しく説明する。図3(a)に各リクエストのサイクル数の例を示し、簡単のため、第1のリクエスト(REQ1)、第2のリクエスト(REQ2)は、1フレーム時間内に3回ずつのアクセスを必要とし、

各リクエストは、定まったワード数のデータのアクセスを必要としている。サイクル数は、共有メモリ側のクロック数とする。各リクエストのサイクル数でアクセスするアドレスによって共有メモリ側のクロック数が変化することがある。SDRAMの場合、同じワード数のデータをアクセスする場合でも、ローアドレスが変更、連続アドレスかどうかに従い、最も少ないサイクル数でアクセスするためには、SDRAMへ必要なアドレス情報や、コマンド、シーケンスに変更が必要となる。第1のリクエスト (REQ1) のAレベルの処理で最大のサイクル数を  $N1A(max)$ 、最も少ないサイクル数を  $N1A(min)$  とし、平均的なサイクル数を  $N1A(ave.)$  として示し、Bレベルの処理で最大のサイクル数を  $N1B(max)$ 、最も少ないサイクル数を  $N1B(min)$  とし、平均的なサイクル数を  $N1B(ave.)$  として示し、 $N1A(max) - N1A(ave.)$ 、 $N1B(max) - N1B(ave.)$  を黒色で示し、 $N1A(ave.) - N1A(min)$ 、 $N1B(ave.) - N1B(min)$  を白色で示している。第2のリクエスト (REQ2) についても同様である。

### 【0033】

図3(b)に、1フレーム内の時間での全サイクル数と、そのサイクル数設計について示している。1フレーム時間内の共有メモリアccessのサイクル数は、 $N1A(max) \times 3 + N2A(max) \times 3$  の場合が最大となる。設計上限界サイクルを図のように設定すると、システムが破綻しないようにリクエストに対して各処理レベルを判定することにより、最終アクセスが終わった時点で限界サイクルを越えないように制御することになる。

### 【0034】

図3(b)の処理レベル制御の例1において、限界サイクル数  $L1$  は、最大サイクル数  $N1A(max) \times 3 + N2A(max) \times 3$  よりも、 $N2A(max) - N2B(max)$  サイクル分少ない、  
 $N1A(max) \times 3 + N2A(max) \times 3 - (N2A(max) - N2B(max)) = N1A(max) \times 3 + N2A(max) \times 2 + N2B(max)$  サイクル、  
 とする。これは、最後のアクセス開始前の矢印Xで示したタイミングで、処理レ



ベルを判定する場合である。残サイクル数が、Aレベルの処理の最大サイクル数  $N1A(max)$  以上であれば、Aレベルの処理を選択し、残サイクル数がAレベルの処理の最大サイクル数  $N1A(max)$  未満であれば、Bレベルの処理を選択する。これにより、最後のアクセス開始前に、消費サイクル数が最大  $N1A(max) \times 3 + N2A(max) \times 2$  サイクルになった場合、最後の処理ではBレベルの処理を選ぶこととなり、このBレベルの処理が最大処理サイクル数となった場合であっても、合計サイクル数は、 $N1A(max) \times 3 + N2A(max) \times 2 + N2B(max)$  サイクル、となり、限界サイクル数  $L1$  を越えることはなく、破綻を防ぐことができる。

### 【0035】

以下、本実施の形態1における、処理レベルの判定方法を一般化する。単位時間で必要な処理を記号で表現し、 $n$  番目 ( $n=1, 2, \dots, N$ ) の処理の  $q(n)$  レベルの処理を、 $pr(n, q(n))$ 、そのサイクル数を、 $C(pr(n, q(n)))$  とする。ここで、サイクル数  $C(pr(n, q(n)))$  の最大サイクル数を、 $C(pr(n, q(n)_{min}))_{max}$  とする。また、 $n$  番目の処理の直前の残りサイクル数を、 $R(n)$  とする。

### 【0036】

処理レベルの判定時、 $n0$  番目の処理  $pr(n0, q(n))$  の直前の、残りサイクル数  $R(n0)$  が、残りの必要な処理をすべて最低レベル（最大のサイクル数が最も小さい処理レベル  $q(n)_{min}$ ）で行うときのアクセスサイクル数の合計、 $\sum_{n=n0}^N C(pr(n, q_{min}))_{max}$ 、以上であれば、この最低レベル  $q_{min}$  を選択することにより破綻は生じない。また、 $n0$  番目の次アクセス処理が、 $q$  レベルの最大サイクル数  $C(pr(n0, q))_{max}$ 、を消費しても、その結果の残りサイクル数、 $R(n0+1) = R(n0) - C(pr(n0, q))_{max}$ 、が、次アクセス後の残りの必要な処理をすべて最低レベル（最大のアクセスサイクルが最小の処理レベル）で行うときのアクセスサイクル数の合計、 $\sum_{n=n0+1}^N C(pr(n, q_{min}))_{max}$ 、以上であれば、この最低レベルを選択することにより破綻は生じない。処理レベル  $q$  は、この条件を満たすレベルを選択できる。

## 【0037】

すなわち、

$$R(n_0) - C(p_r(n_0, q(n_0)))_{\max}$$

$$\geq \sum_{n=n_0+1}^N C(p_r(n, q(n)_{\min}))_{\max}$$

となる処理レベル  $q$  を、選ぶことができる。

## 【0038】

尚、初期状態で、 $R(1) = L$  (限界サイクル数)、かつ、

$$R(1) - C(p_r(1, q(1)))_{\max}$$

$$\geq \sum_{n=2}^N C(p_r(n, q(n)_{\min}))_{\max}$$

となって、破綻しない処理レベルが存在するためには、

$$R(1) = L$$

$$\geq C(p_r(1, q(1)))_{\max} + \sum_{n=2}^N C(p_r(n, q(n)_{\min}))_{\max}$$

$$\geq C(p_r(1, q(1)_{\min}))_{\max} + \sum_{n=2}^N C(p_r(n, q(n)_{\min}))_{\max}$$

ax

$$= \sum_{n=1}^N C(p_r(n, q(n)_{\min}))_{\max}$$

すなわち、

$$L \geq \sum_{n=1}^N C(p_r(n, q(n)_{\min}))_{\max}$$

である、ことが必要である。

## 【0039】

図3(b)の限界サイクル  $L_2$  の場合の例を示す。

$$R(n_0) - C(p_r(n_0, q(n_0)))_{\max}$$

$$\geq \sum_{n=n_0+1}^N C(p_r(n, q(n)_{\min}))_{\max}$$

で、 $n_0 = 1, 2, 3$  では、 $q(n_0)$  として A レベルを選択できるが、 $n_0 = 4$  では、

$$R(4) - C(p_r(4, q(4)))_{\max}$$

$$\geq \sum_{n=5}^6 C(p_r(n, q(n)_{\min}))_{\max}$$

となる  $q(4)$  として、B レベルしか選べない。これは 4 番目 ( $n_0 = 4$ ) のアクセスで A レベルを指示した場合、結果が A レベルの  $\max$  であると、5 番目、6 番目のアクセスで B レベルを指示しても、それぞれが  $\max$  であると破綻する

ので、4番目のアクセスについてBレベルを指示することに相当する。5番目のアクセス、6番目のアクセス、についても同様である。

#### 【0040】

このように、本実施の形態1によるバスコントローラにおいては、アクセスサイクル数カウンタ12と、リクエスト毎に異なる処理レベルの処理を行う処理手段であるアービタ11と、各リクエストの異なる処理レベルの処理とアクセスサイクル数との対応を記述した対応表13とを備え、上記アービタ11は、複数のリクエストからのアクセス要求に対応してアクセスの許可を調停するとともに、その処理レベルを指定する処理レベル判定手段としても機能するものとしたので、アービタ11により、現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、対応表とに従い、共有メモリ21へのアクセスをどのリクエストに許可するかを決定するとともに、該アクセスを許可したリクエストの処理レベルを指定するようにし、該処理レベルを指定するにおいて限界サイクル数を越えることが予測される場合には、サイクル数の少ない処理レベルを選択することにより、全リクエストについての、最大アクセスサイクル数×最大アクセス回数の総和よりも小さいサイクル数で設計することが可能となる。たとえば、リクエストの2個後の処理、すなわち、現在アクセスが完了したリクエストの次のリクエストのさらに次のリクエストのアクセス処理、から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときの、アクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で次のリクエストの処理のレベルを選択する、ようにすることにより、適切な設計を行うことが可能となる。

#### 【0041】

したがって、高い周波数によるバスシステム設計によるコストアップを抑えることができ、かつ、最大サイクル数が増加した場合でも、最大サイクル数よりも小さいサイクル数、すなわち低い周波数でバスコントローラの設計を行うことができ、再設計を抑えることができる効果がある。

#### 【0042】

(実施の形態 2)

次に、本発明の実施の形態 2 について、説明する。

従来例と同様、図 6 のバスシステムの例を用いて説明する。

第 1 のリクエスト 63 と、第 2 のリクエスト 64 は、リアルタイム系の処理であり、画像 1 フレームの時間にそれぞれ 2 種類の処理レベルを持ち、それぞれの処理レベルに従った定まった量の共有メモリへのアクセス、及びデータ処理を必要とする。2 種類の処理レベルで共有メモリのアクセスサイクル数が多い方を A レベル、少ない方を B レベルとする。第 1 のリクエスト 63 は、画像のフィルタリング処理であり、A レベルとして現フレームと前フレームのデータを用いたフィルタリング処理、B レベルとして現フレームのデータを用いたフィルタリング処理を、行うものとする。第 2 のリクエスト 64 は、画像符号化処理であり、A レベルとして広い動きベクトル検出範囲用の参照画像を用いた画像符号化処理を、B レベルとして、狭い動きベクトル検出範囲用の参照画像を用いた画像符号化処理を、行うものとする。両方のリクエストとも、A レベルの方が画質は良くなるが、メモリサイクル数は多くなるものとする。第 3 のリクエスト 65 は、非リアルタイム系の処理であり、画像の処理に比較して非定期的な共有メモリへのアクセス、及びデータ処理を行なう。

【0043】

バスコントローラ 62 の構成は、図 1 と同様である。

サイクルカウンタ 12 は、メモリプロトコル変換手段 14 からの情報に従い、共有メモリのアクセスサイクル数をカウントし、アクセス毎に現時点のサイクル数を出力する。

【0044】

アービタ 11 は、現時点のサイクル数と、予め定められた限界サイクル数とから残りサイクル数を計算し、リクエスト処理レベルーサイクル数対応表 13 の情報を用いて、優先順位に従ってリクエストの内の一個に許可信号と処理レベル制御信号を与える。第 1 のリクエスト、第 2 のリクエストの順の繰り返りでアクセス要求が入力され、これらを順に許可し、第 3 のリクエストの非定期的要求に対して、残サイクル数と、リクエスト処理レベルーサイクル数対応情報に従って、

許可の挿入を行なう。

#### 【0045】

メモリプロトコル変換手段14は、リクエストのアクセスに用いられるアドレス、データ、READかWRITEを示すリクエストアクセス信号を、共有メモリアクセス信号にプロトコル変換するとともに、プロトコル変換やアクセス速度の変換のためのバッファ機能を持つ。共有メモリアクセス信号としては、SDRAMアクセスのためのRowアドレス、Columnアドレス、コマンド発行などがある。

#### 【0046】

第1リクエスト63、第2リクエスト64、第3リクエスト65は、各リクエスト内の処理に従い、バスコントローラ62に共有メモリ61へのアクセス要求を出し、バスコントローラ62の許可と、処理レベル制御信号に従って、バスコントローラ62のプロトコル変換を通して、共有メモリ61にアクセスを行なう。

#### 【0047】

図4のサイクル数の例で動作を詳しく説明する。図4(a)に各リクエストのサイクル数の例を示し、簡単のため、第1のリクエスト(REQ1)、第2のリクエスト(REQ2)は、1フレーム時間内に2回ずつのアクセスを必要とし、第3のリクエスト(REQ3)は、1フレーム時間内に0回または1回のアクセスを必要とする。各リクエストは、定まったワード数のデータのアクセスを必要としている。サイクル数は、共有メモリ側のクロック数とする。各リクエストのサイクル数でアクセスするアドレスによって、共有メモリ側のクロック数が変化することがある。SDRAMの場合、同じワード数のデータをアクセスする場合でも、ローアドレスが変更、連続アドレスかどうかに従い、最も少ないサイクル数でアクセスするためには、SDRAMへ必要なアドレス情報や、コマンド、シーケンスに変更が必要となる。第1のリクエスト(REQ1)のAレベルの処理で最大のサイクル数をN1A(max)、最も少ないサイクル数をN1A(min)とし、平均的なサイクル数をN1A(ave.)として示し、Bレベルの処理で最大のサイクル数をN1B(max)、最も少ないサイクル数をN1B(m

in)とし、平均的なサイクル数を $N1B(ave.)$ として示し、 $N1A(max) - N1A(ave.)$ 、 $N1B(max) - N1B(ave.)$ を黒色で示し、 $N1A(ave.) - N1A(min)$ 、 $N1B(ave.) - N1B(min)$ を白色で示している。第2のリクエスト(REQ2)についても同様である。第3のリクエスト(REQ3)については、処理レベルは1種類で、最大のサイクル数を $N3(max)$ 、最も少ないサイクル数を $N3(min)$ とし、平均的なサイクル数を $N3(ave.)$ として示し、 $N3(max) - N3(ave.)$ 、 $N3(ave.) - N3(min)$ を黒色で示し、 $N3(ave.) - N3(min)$ を白色で示している。

#### 【0048】

図4(b)で1フレーム内の時間での全サイクル数と、そのサイクル数設計について示している。1フレーム時間内の共有メモリアクセスのサイクル数は、 $N1A(max) \times 2 + N2A(max) \times 2 + N3(max)$ の場合が最大となる。設計上限界サイクルを図のように設定すると、システムが破綻しないようにリクエストに対して各処理レベルを判定することにより、最終アクセスが終わった時点で限界サイクルを越えないように制御することになる。

#### 【0049】

図4(b)の限界サイクル数 $L1$ の動作例において、限界サイクル数 $L1$ が最大サイクル数 $N1A(max) + N2A(max) + N3(max) + N1B(max) + N1B(max)$ サイクルとする。2番目のアクセス後の矢印Xで示したタイミングまでに第3リクエストからの要求があり、処理レベルを判定する場合の動作を考える。

#### 【0050】

残サイクル数が、 $N3(max) + N1B(max) + N1B(max)$ サイクル以上、であれば、第3のリクエストの要求に許可を与える。この例では、第3のリクエストの3番目のアクセスが最大サイクル数の場合でも、1フレーム内の残りのアクセスである、1回の第1リクエストのアクセス、1回の第2リクエストのアクセス、をBレベルとすることにより、破綻せずにアクセスを完了することができる。

## 【0051】

図4 (b) の限界サイクル数  $L2$  の動作例において、限界サイクル数  $L2$  が最大サイクル数  $N1A(max) + N2A(max) + N3(max) + N1B(max) + N1B(max)$  サイクルよりも少ないとする。2番目のアクセス後の矢印Xで示したタイミングまでに、第3リクエストからの要求があり、処理レベルを判定する場合の動作を考える。

## 【0052】

残サイクル数が、 $N3(max) + N1B(max) + N1B(max)$  サイクル以上、であれば、第3のリクエストの要求に許可を与える。この例では第3のリクエストのアクセスを許可してしまうと、1フレーム内の残りのアクセスである、1回の第1リクエストのアクセス、1回の第2リクエストのアクセス、をBレベルとしても、最大のサイクル数であれば破綻してしまう。したがって、ここでは破綻をさけるために、第3リクエストのアクセス要求に対しては許可を与えない。代わりに第2フレームの最初に第3のリクエストによるアクセスを許可することなどにより、待たされるサイクル数を有効に抑えることもできる。

## 【0053】

処理レベル判定を一般化する。単位時間で必要な処理を実施の形態1と同様に記号で表現し、非リアルタイムの処理をSとする。この処理の最大サイクル数  $C(S)_{max}$  を消費しても、その結果の残りサイクル数  $R(n0) - C(S)_{max}$  が、次アクセス後の残りの必要な処理をすべて最低レベル（最大のアクセスサイクルが最小の処理レベル）で行うときのアクセスサイクル数の合計、 $\sum_{n=n0}^N C(pr(n, q_m(n)_{in}))_{max}$  以上、であれば、最低レベルを選択することにより破綻は生じない。従って、処理Sを許可することができる。

## 【0054】

すなわち、

$$\begin{aligned} & R(n0) - C(S)_{max} \\ & \geq \sum_{n=n0}^N C(pr(n, q(n)_{min}))_{max} \end{aligned}$$

となる処理Sを許可することができる。

## 【0055】

リアルタイム処理と同様に、非リアルタイム処理についても処理レベルを設定できる場合には、 $q$  レベルの非リアルタイム処理を  $S(q)$  として、

$$R(n_0) - C(S(q)) \max \\ \geq \sum_{n=n_0}^N C(p_r(n, q(n) \min)) \max$$

となる  $q$  レベルの処理  $S(q)$  を、許可することができる。

#### 【0056】

このように、本実施の形態 2 によるバスコントローラにおいては、アクセスサイクル数カウンタ 12 と、リクエスト毎に異なる処理レベルの処理を行う処理手段であるアービタ 11 と、各リクエストの異なる処理レベルの処理とアクセスサイクル数との対応を記述した対応表 13 とを備え、上記アービタ 11 は、複数のリクエストからのアクセス要求に対応してアクセスの許可を調停するとともに、その処理レベルを指定する処理レベル判定手段としても機能するものとしたので、アービタ 11 により、現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、対応表とに従い、限界サイクル数を越えることが予測される場合には、非リアルタイム系のバスアクセス要求を許可しない制御をするようにすることにより、全リクエストについての、最大アクセスサイクル数×最大アクセス回数の総和よりも小さいサイクル数で設計することが可能となる。たとえば、次のリクエストである非リアルタイム系リクエストの次の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で処理を完了できない場合は、次の非リアルタイム系のリクエストを許可しない、ようにすることにより、非リアルタイム系の処理サイクル数を含めない総処理サイクル数で設計を行うことが可能となる。

#### 【0057】

したがって、高い周波数によるバスシステム設計によるコストアップを抑えることができ、かつ、最大サイクル数が増加した場合でも、最大サイクル数よりも小さいサイクル数、すなわち低い周波数でバスコントローラの設計を行うことができ、再設計を抑えることができる効果がある。



## 【0058】

## 【発明の効果】

以上のように、本発明の請求項1の発明にかかるバスコントローラによれば、共有メモリに対しアクセスする複数のリクエストの処理レベルを制御するバスコントローラであって、上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、上記各リクエスト毎に異なる処理レベルの処理を行う処理手段と、上記各リクエスト毎に異なる処理レベルの処理と上記アクセスサイクル数との対応を記述した対応表と、アクセスが許可される各リクエストの処理手段による処理レベルを指定する処理レベル判定手段と、を備え、上記処理レベル判定手段は、上記アクセスサイクル数カウンタによりカウントされる現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記各リクエスト毎に異なる処理レベルの処理と上記アクセスサイクル数との対応表とに従い、アクセスが許可される各リクエストの処理レベルを指定する、ものとしたので、全リクエストのサイクル数の合計が、限界サイクル数を越えることが予測される場合には、サイクル数の少ない処理レベルを選択することにより、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数とを用いて設計することを可能とし、サイクル上の破綻を生じさせずに、設計コスト、設計の困難さを抑えることができる効果が得られる。

## 【0059】

本発明の請求項2の発明によれば、共有メモリに対しアクセスする複数のリクエストからのアクセス要求に対応して調停するバスコントローラであって、上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、上記複数のリクエストとアクセスサイクル数との対応表と、上記複数のリクエストからのアクセス要求に対してアクセスの許可を調停するアービタと、を備え、上記アービタは、上記アクセスサイクル数カウンタによりカウントされる現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記複数のリクエストとアクセスサイクル数との対応表とに従い、全リクエストのサイクル数の合計が、限界サイクル数を越えることが予測される場合には

、非リアルタイム系のバスアクセス要求を許可しない制御をする、ものとしたので、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数とを用いて設計することを可能とし、サイクル上の破綻を生じさせずに、設計コスト、設計の困難さを抑えることができる効果が得られる。

#### 【0060】

本発明の請求項3の発明によれば、共有メモリに対しアクセスする複数のリクエストの処理レベルを制御し、複数のリクエストからのアクセス要求に対して調停するバスコントローラであって、上記共有メモリをアクセスするアクセスサイクル数をカウントするアクセスサイクル数カウンタと、上記各リクエスト毎に異なる処理レベルの処理を行う処理手段と、上記各リクエスト毎に異なる処理レベルの処理とアクセスサイクル数との対応を記述した対応表と、アクセスが許可される各リクエストの処理手段による処理レベルを指定する処理レベル判定手段と、上記複数のリクエストからのアクセス要求に対応しアクセスの許可を調停するアービタと、を備え、上記処理レベル判定手段、および上記アービタは、上記アクセスサイクル数カウンタによりカウントされる現サイクル数と、予め定められた限界サイクル数に対する残サイクル数と、上記対応表とに従い、アクセスが許可されるリクエストの処理手段による処理レベルを指定するとともに、全リクエストのサイクル数の合計が、限界サイクル数を越えることが予測される場合には、非リアルタイム系のバスアクセス要求を許可しない制御をする、ものとしたので、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数を用いて設計することが可能となり、サイクル上の破綻を生じさせずに、設計コスト、設計の困難さを抑えることができる効果が得られる。

#### 【0061】

本発明の請求項4の発明によれば、請求項1記載のバスコントローラにおいて、上記処理レベル判定手段は、リクエストの2個後の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数

を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で次のリクエストの処理のレベルを選択するものとしたので、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数とを用いて設計することを可能となり、サイクル上の破綻を生じさせずに、設計コスト、設計の困難さを抑えることができ、しかも、ほとんど全期間にわたって最適な処理を行うことができる効果がある。

#### 【0062】

本発明の請求項5の発明によれば、請求項2記載のバスコントローラにおいて、上記アービタは、次のリクエストである非リアルタイム系リクエストの次の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で処理を完了できない場合は次の非リアルタイム系のリクエストを許可しない、ようにすることとしたので、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数とを用いて設計することを可能とし、サイクル上の破綻を生じさせずに、設計コスト、設計の困難さを抑えることができ、しかも、ほとんど全期間にわたって最適な処理を行うことができる効果がある。

#### 【0063】

本発明の請求項6の発明によれば、請求項3記載のバスコントローラにおいて、上記処理レベル判定手段、および上記アービタは、次のリクエストである非リアルタイム系リクエストの次の処理から基準時間の最後の処理までの処理を、最大のサイクル数が最も小さい処理レベルで行うときのアクセスサイクル数の総和を計算し、限界サイクル数から現在のアクセスサイクル数を減算して残サイクル数を計算し、上記残サイクル数から上記総和を減算したサイクル数までの範囲で処理を完了できない場合は、次の非リアルタイム系のリクエストを許可しない、ものとしたので、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数とを用いて設計することが可能となり、サイクル上の破綻を生じさせずに、設計コスト、設計の困難さを抑えることができ

、しかも、ほとんど全期間にわたって最適な処理を行うことができる効果がある。

#### 【0064】

本発明の請求項7の発明によれば、請求項6記載のバスコントローラにおいて、上記非リアルタイム系リクエストは、複数の異なる処理レベルを有するもの、としたので、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数とを用いて設計することが可能となり、サイクル上の破綻を生じさせずに、設計コスト、設計の困難さを抑えることができ、しかも、より一層、ほとんど全期間にわたって最適な処理を行うことができる効果がある。

#### 【図面の簡単な説明】

##### 【図1】

本発明の実施の形態1によるバスコントローラの構成図

##### 【図2】

上記実施の形態1における、バスシステムの構成例を示す図

##### 【図3】

上記実施の形態1における、第1の処理サイクルを説明する図

##### 【図4】

上記実施の形態1における、第2の処理サイクルを説明する図

##### 【図5】

従来のバスコントローラの構成図

##### 【図6】

従来の、バスシステムの構成例を示す図

##### 【図7】

従来の処理サイクルを示す図

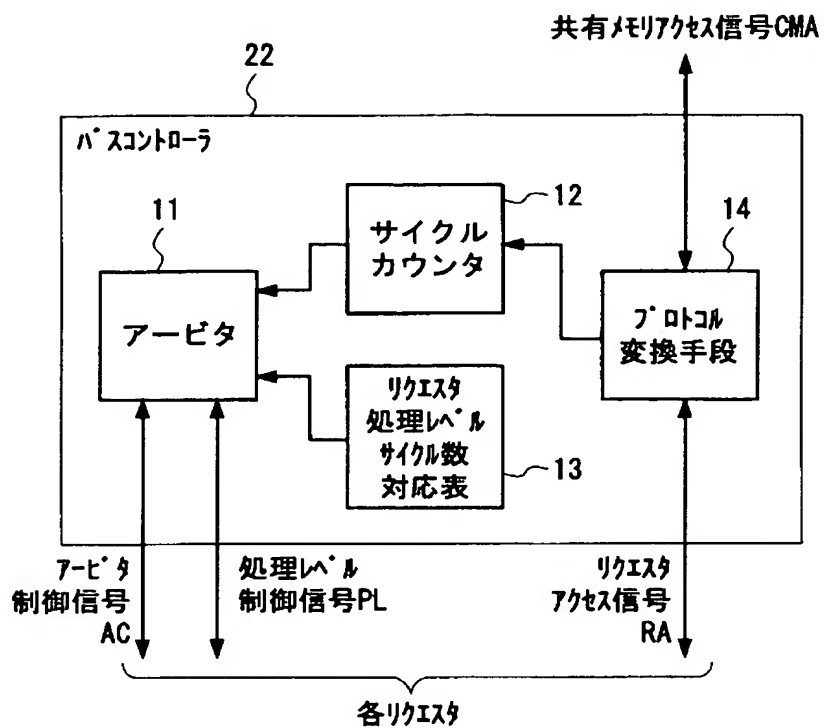
#### 【符号の説明】

- 11     アービタ
- 12     サイクルカウンタ
- 13     リクエスト処理レベルーメモリサイクル数対応表

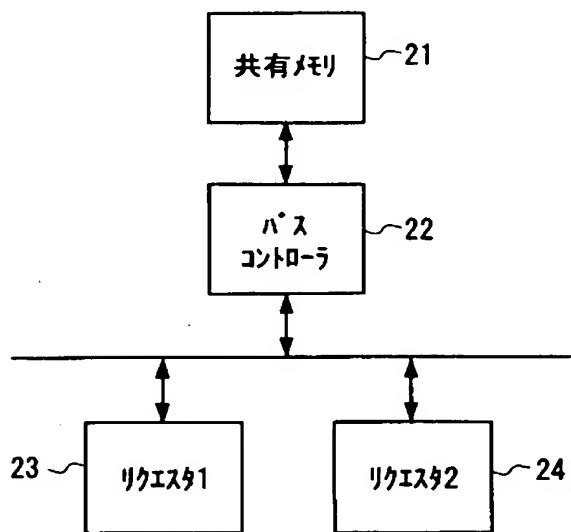
- 1 4     プロトコル変換手段
- 2 1     共有メモリ
- 2 2     バスコントローラ
- 2 3, 2 4   リクエスタ
- A C     アービタ制御信号
- P L     処理レベル制御信号
- R A     リクエスタアクセス信号
- CMA   共有メモリアクセス信号

【書類名】 図面

【図 1】

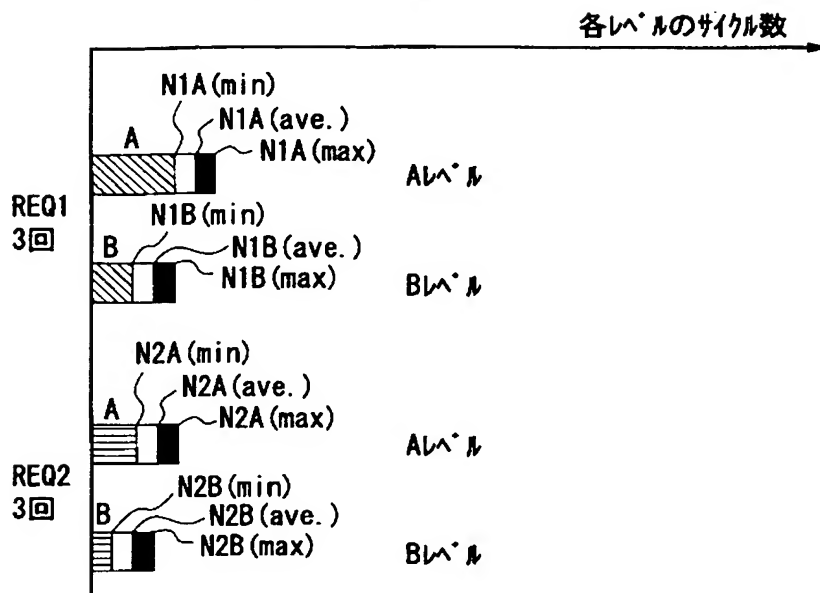


【図 2】

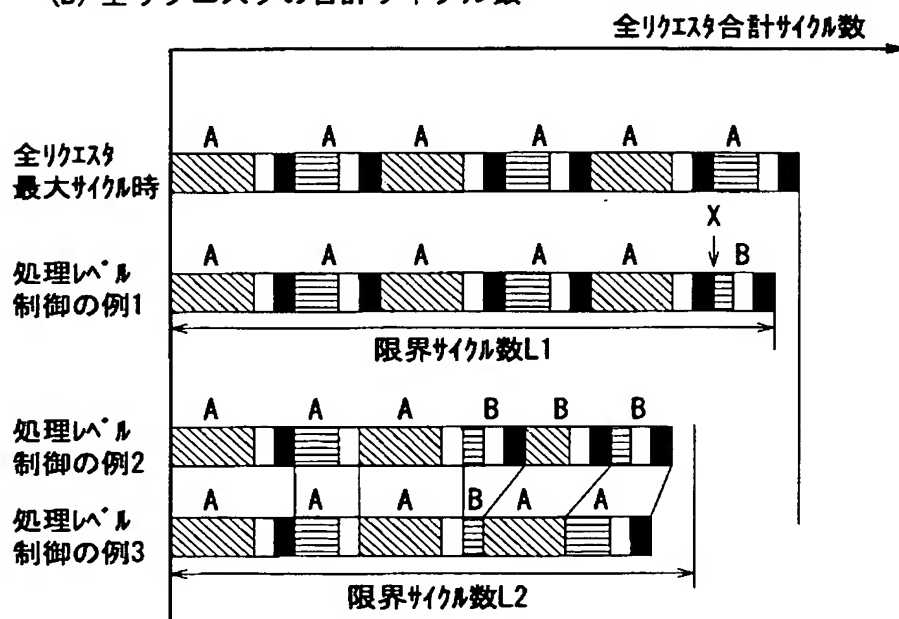


【図 3】

(a) 各リクエストの最大サイクル数

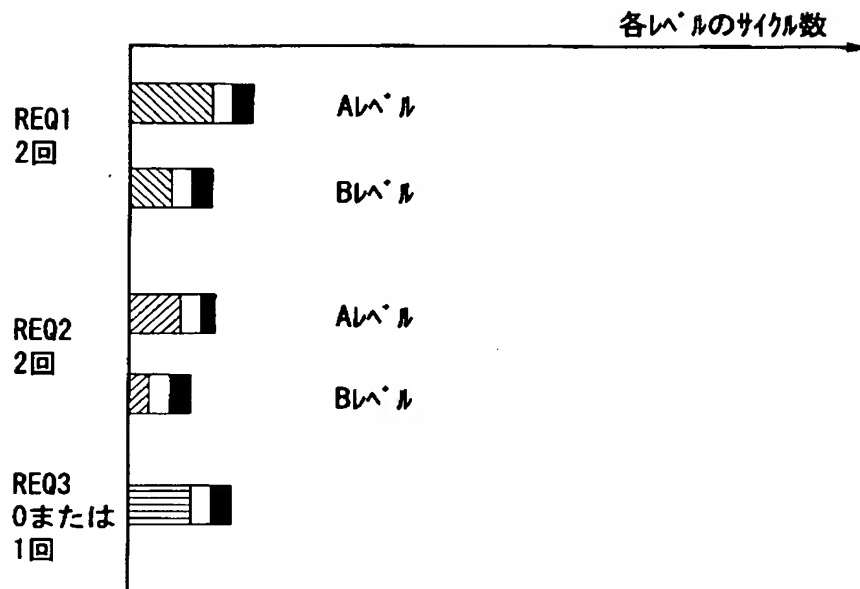


(b) 全リクエストの合計サイクル数

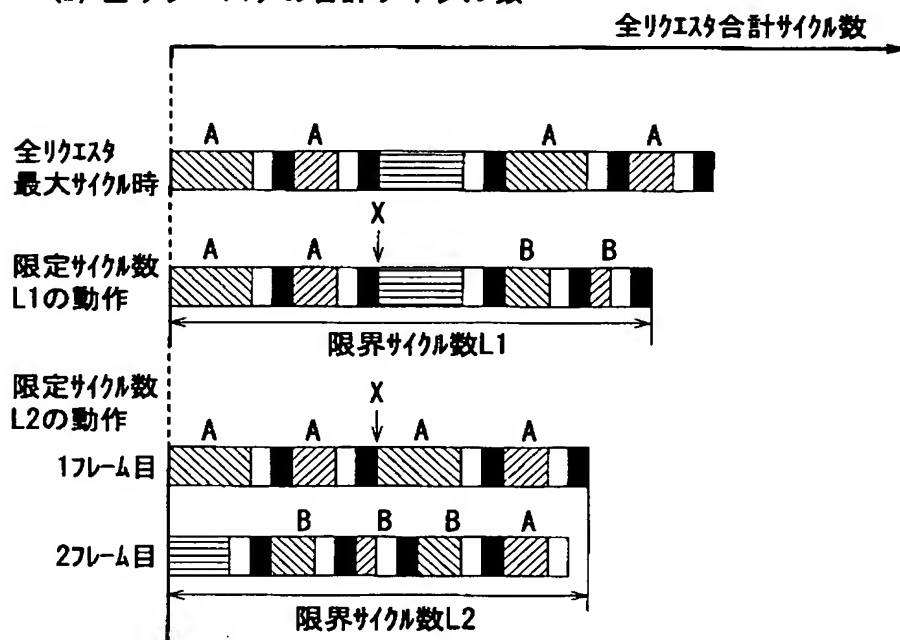


【図 4】

(a) 各リクエストの最大サイクル数

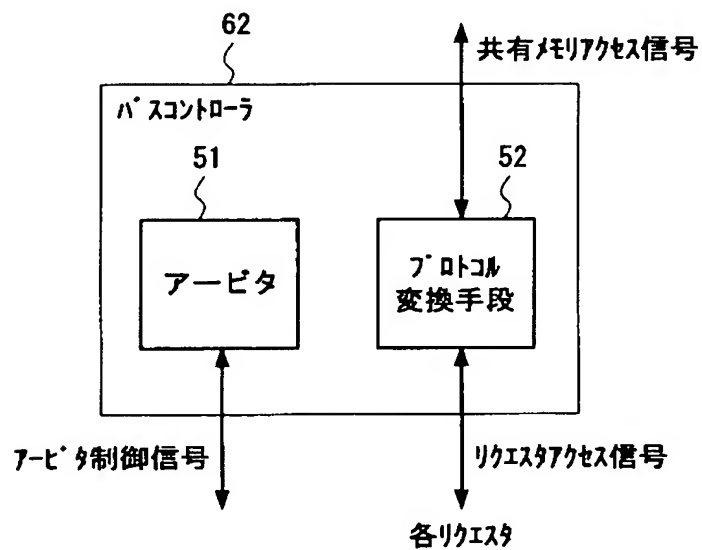


(b) 全リクエストの合計サイクル数

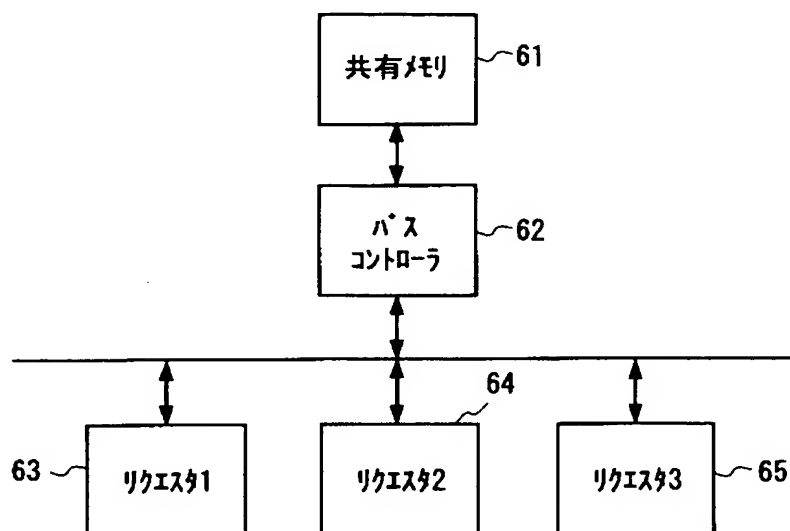




【図 5】

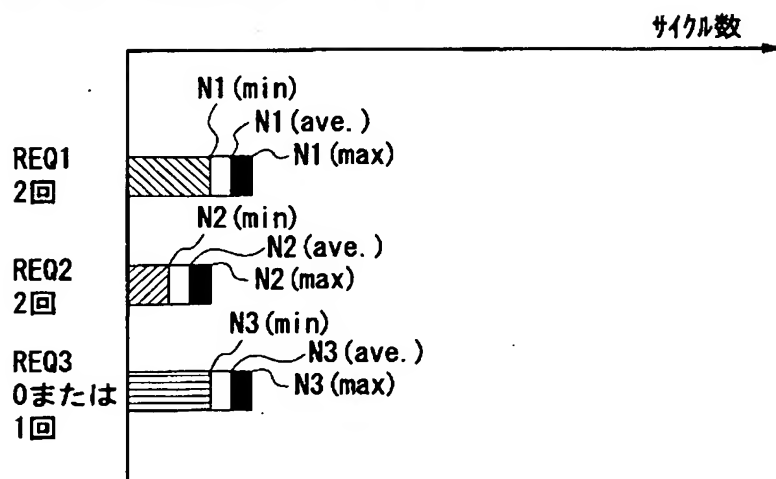


【図 6】

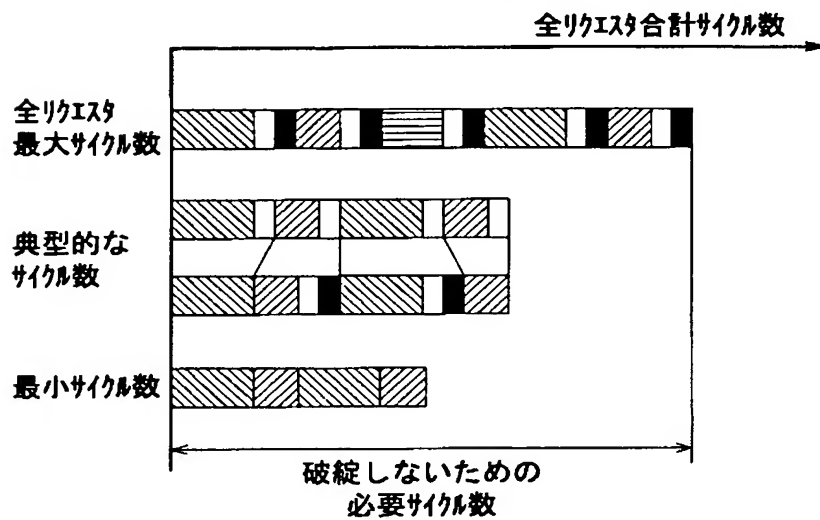


【図 7】

(a) 各リクエストのサイクル数



(b) 全リクエストの合計サイクル数



【書類名】 要約書

【要約】

【課題】 バスコントローラは最大のサイクル数で設計を行なうと、周波数が高く、設計困難度が高く、コストアップとなる

【解決手段】 共有メモリ 2 1 に対してアクセス要求を行う、各リクエスト 2 3 , 2 4、あるいは 6 3, 6 4, 6 5 毎に、サイクル数の異なる処理レベルの処理手段を備えて、現サイクル数に従って限界サイクル数を越えることが予測される場合に、サイクル数の少ない処理レベルを選択し、あるいは、非リアルタイム系のバスアクセス要求を許可しない制御をして、全リクエストの最大アクセスサイクル数と、その最大アクセス回数分の総和よりも小さいサイクル数で、設計することを可能とした。

【選択図】 図 1

特願 2 0 0 3 - 0 7 6 0 5 5

出 願 人 履 歷 情 報

識別番号

[ 0 0 0 0 0 5 8 2 1 ]

1. 変更年月日

1 9 9 0 年 8 月 2 8 日

[変更理由]

新規登録

住 所

大阪府門真市大字門真 1 0 0 6 番地

氏 名

松下電器産業株式会社